Akeem Kennedy/Zefeng Zhang/Abhishek Bhandari
ECE110 Honors Lab
12/3/16

Honor Project Lab Report

Introduction

Our project is to build a remote that could control the window shades and the light switch in our rooms. We have high-loft beds and it is a bit of a problem to manually switch the light off before we go to sleep. In the morning, the penetrating sunlight can be so discomforting and climbing out the bed and changing the window shades is inconvenient. The remote control can pull down the shades when required and change its orientation such that it is at an oblique angle to the sunlight.

Analysis of Components

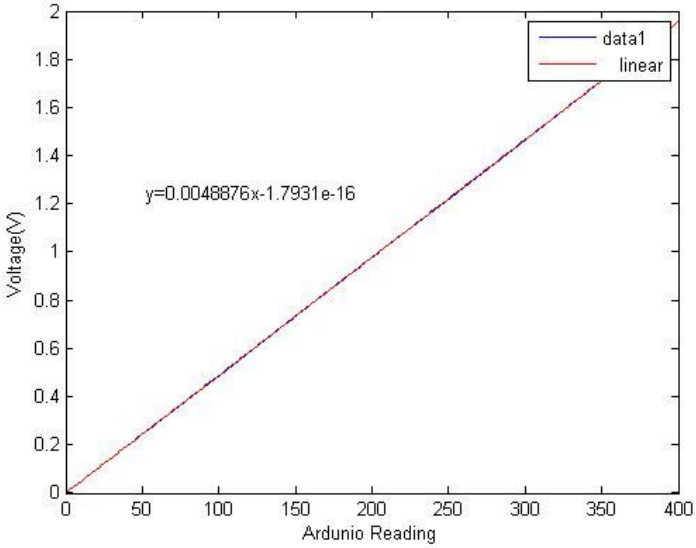
Infrared Sensor Characterization(Description)

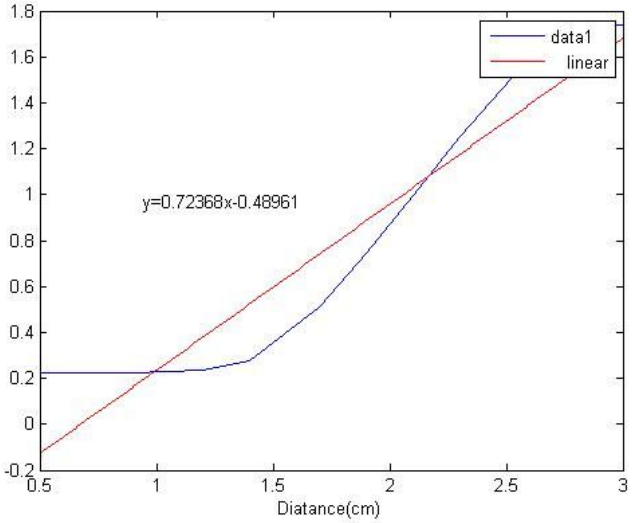
The infrared emitter and detector are operate at 940nm and work well for generic IR systems including remote control and touch-less object sensing. Using a simple ADC on any microcontroller will allow variable readings to be collected from the detector. The emitter is driven up to 50mA with a current limiting resistor as with any LED device. The detect is a NPN transistor that is biased by incoming IR light.

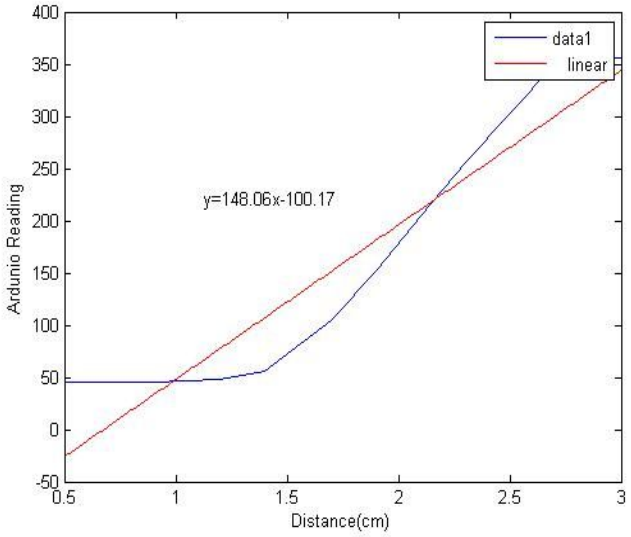
Infrared Sensor Characterization(Values)

Distance(cm)	*Arduino Reading*	*Voltage(V)*	*Comment*
0.5	*46*	*0.2248*	
0.9	*46*	*0.2248*	
1.2	*48*	*0.2346*	
1.4	*56*	*0.2737*	
1.7	*105*	*0.5132*	
1.9	*152*	*0.7429*	*When the LED light up*
2.1	*204*	*0.9971*	
2.3	*256*	*1.2512*	
2.7	*350*	*1.7107*	

3	*356*	*1.74*	

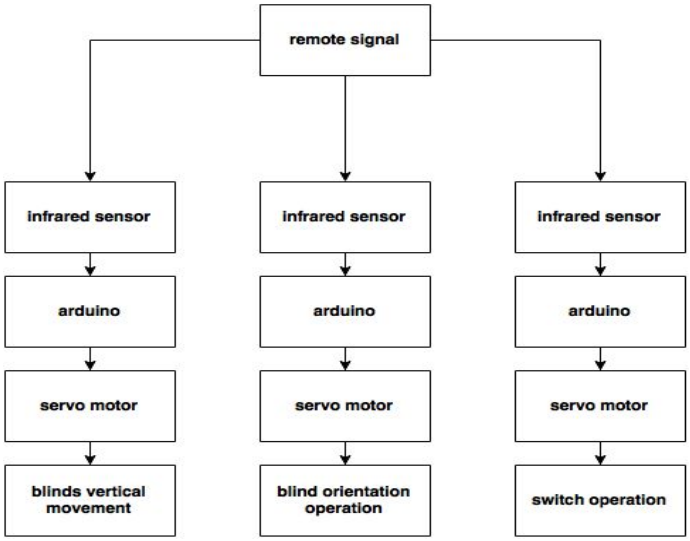

data1
linear
y=0.0048876x-1.7931e-16
Voltage(V)
Ardunio Reading


data1
linear
y=0.72368x-0.48961
Diatance(cm)

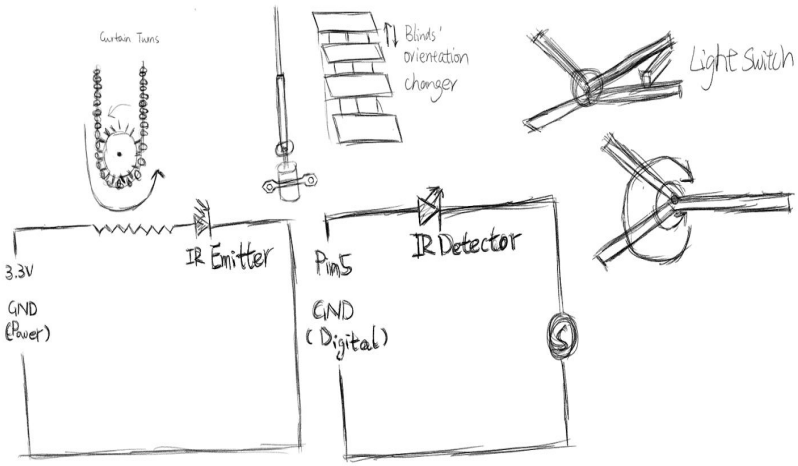

data1
linear
y=148.06x-100.17
Ardunio Reading
Distance(cm)

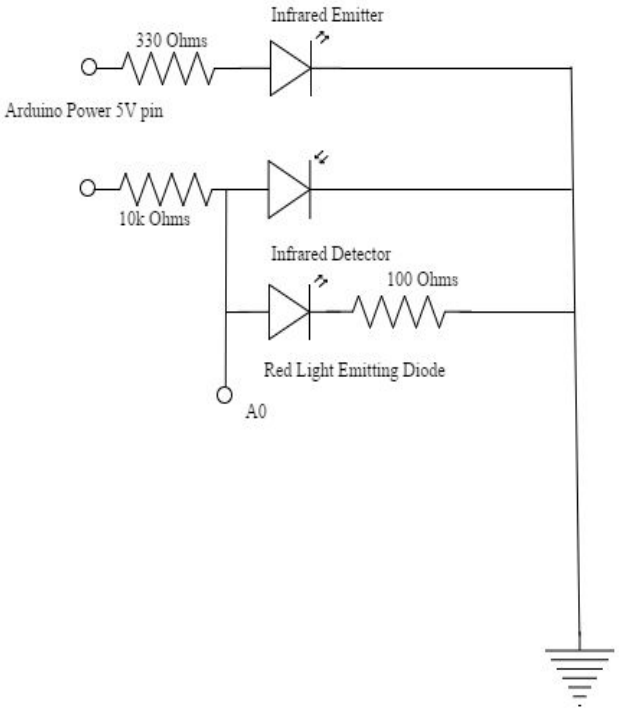
Design Description

Block Diagram



Circuit Schematic



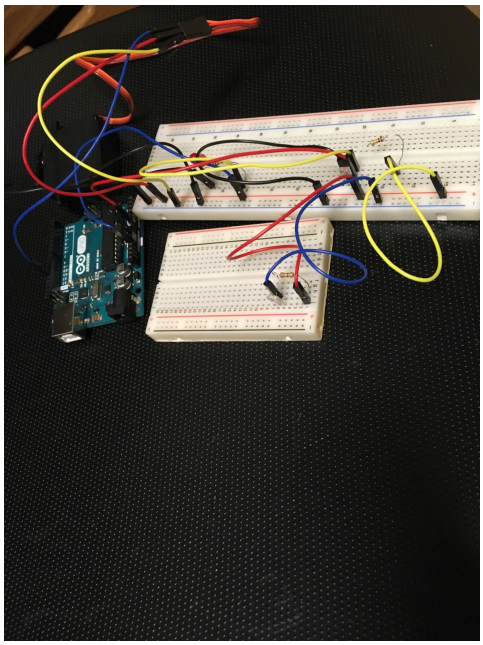


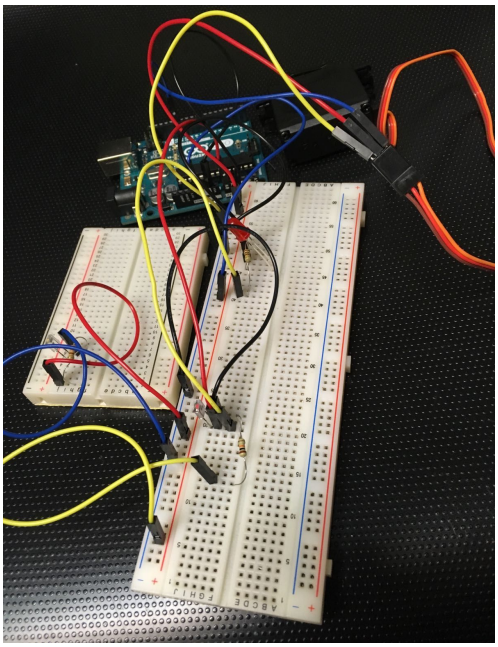
Description of Circuit Design

The circuit was constructed to efficiently allow the user to remotely switch the light on and off. The infrared sensors were placed adjacent to each other to limit interference from outside sources. Then the digital read came from the output of the detector to the arduino's analog pin.

Physical/mechanical construction

The servo motor was attached to the same board as the detector so that the output values from the detector would tell the servo how to move. The servo was then placed on the wall or block in our case to simulate how it would turn the light switch on and off.





Conclusion

Lessons Learned: One problem we faced was troubleshooting our circuit when problems occurred. We always referred back to the schematic and circuit diagram for assistance but our analysis would come up that there was nothing wrong. One problem was that our servo motor was put in wrong but after correcting this our basic concept worked. But the biggest problem that we encounter was when it was time to code. Since all member of our team is in ECE 110 we did not have any familiarity with coding but we worked through it with the help of some advisers and a few tutorials.

Self-Assessment : According to the goal of the lab, we were able to reach the goal. We wanted our device to turn the light switch on and off and it does. In theory we could apply the same concept to the blinds and it should also work. This is what we wanted to accomplish and I think we did.

Appendix

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // twelve servo objects can be created on most boards
int pos = 90;    // variable to store the servo position

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
myservo.attach(9);  // attach servo to pin 9
}

// the loop routine runs over and over again forever:
void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
  if (sensorValue > 250)
{ myservo.write(135);
       delay(15); }
  else  {                          // in steps of 1 degree
     myservo.write(45);              // tell servo to go to position in variable 'pos'
     delay(15);                       // waits 15ms for the servo to reach the position
    }
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```